System-Level Service Assurance — The H∀Mcast Approach to Global Multicast*

Thomas C. Schmidt[1] and Matthias Wählisch[2,3]

[1] HAW Hamburg, Department Informatik, Berliner Tor 7, 20099 Hamburg, Germany
[2] link-lab, Hönower Str. 35, 10318 Berlin, Germany
[3] Freie Universität Berlin, Inst. für Informatik, Takustr. 9, 14195 Berlin, Germany
{t.schmidt,waehlisch}@ieee.org

Abstract. The Internet revolution introduced a single, adaptive abstraction layer for global communication. Today, IP interconnects millions of applications, which themselves are bound to the present IP layer via the socket API. After almost 30 years, the time has come to abandon this focus on a single, homogeneously established Internet protocol and thereby release the accumulated needs for innovation on the network layer. H∀Mcast addresses this goal by following the *evolutionary* approach of a hybrid multiservice network layer that decouples service and application development from infrastructure deployment. The objective of this work is a universal, robust service access that allows group applications to run everywhere, no matter what the status of regional technological deployment will be.

Keywords: Internet service architecture, hybrid multicast, multicast mobility management, multicast security.

1 Introduction

Emerging mass applications like IPTV and Massive Multiplayer Online Role Games (MMORGs), but also traditional communication systems such as videoconferencing or newscasts, distribute content to large receiver groups. The most efficient way for group dissemination follows (optimal) distribution trees and is executed on the lowest possible layer [1] available in the network. IP Layer Multicast [2] achieves this goal by providing a corresponding extension of the network socket API and by a dynamic mapping onto the MAC layer. However, a transparent employment of group communication requires a global deployment of IP multicast. Despite of its long-term availability in protocols and implementations, this has not been seen over the years.

The H∀Mcast approach selects the example of group communication to demonstrate how a future multiservice Internet can immediately enable new services,

* This work is supported by the German Federal Ministry of Education and Research. H∀Mcast (http://hamcast.realmv6.org) is part of G-Lab (http://www.german-lab.de).

T. Magedanz et al. (Eds.): TridentCom 2010, LNICST 46, pp. 635–639, 2011.

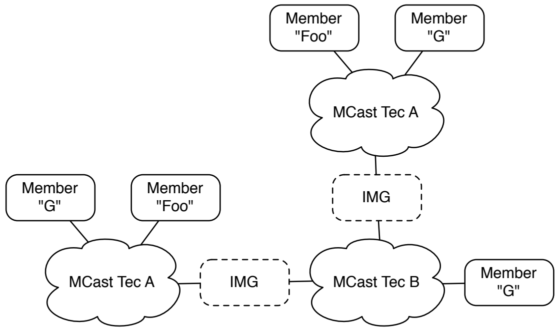

Fig. 1. Reference scenario for hybrid multicast, interconnecting group members from isolated homogeneous and heterogeneous domains

even at a stage of gradual, incomplete deployment. Multicast is to be realized in a concept of multiple, simultaneously operating layers. This will give rise to a continuous *Service Availability*, at first. At second, an adaptive selection will ensure that only the lowest available layer comes into operation and thereby optimize *Service Efficiency*. These functions, combined with further capabilities at end systems, will be hidden behind a uniform programming interface, such that any requirement on network-specific choices or arrangements is taken away from application programmers. A straight forward way to implementing universally deployable products is thus regained.

This paper presents a brief overview of the ongoing project work. In the following section 2, we discuss the core problems and give a conceptual overview of the H∀Mcast system-centric service model. Section 3 introduces the implementation and a new programming interface for group services. A conclusion and an outlook complete this writing.

2 Approaching the Internet Service Problem

During the last decade, the Internet has showed resistance towards deploying new services or technologies, and multicast serves as an excellent example for the underlying deadlock: As group communication is a complex, composite service that can be realized in many flavors and various technologies, individual stakeholders of the Internet community make choices of service design and technological deployment. Since there is no common interface to access these various techniques, applications drive individually implemented solutions that minimize interaction with the network infrastructure and thus act as disincentive towards a universal deployment. Users, programmers and operators not only pay for this with higher efforts, lower efficiency and lack of innovation, but new service requirements such as mobility [3], multi-homing, and security challenge original design parts of the Internet and put additional stress on any rag-type service architecture.

The many flavours and pluralistic technologies of group communication – including IPv4 and IPv6 – require hybrid solutions. As displayed in Figure 1, such approaches need to integrate (1) Multicast domains running the same multicast technology but remaining isolated, possibly only connected by network layer unicast; and (2) Multicast domains running different multicast technologies, but hosting nodes that are members of the same multicast group. Hybrid multicast may be realized with limited efforts and acceptable performance [4,5], but requires enhanced intelligence when accessing the network layer.

Following the end-to-end design principle [1] and inspired by current overprovisioning at end nodes, H∀Mcast allocates this service intelligence at edge systems. The system-centric group communication stack is located in an adaptive, modular middleware that represents an abstraction layer between applications and transport technologies. The middleware is a unique daemon process instantiated once per host at start-up, and includes modules for *service discovery, name-to-address mapping,* and *technology-specific service interfaces*. It provides efficient multicast access for any group application without reimplementation and redundancy, and offers a path to replace current proprietary workarounds deployed in manifold ways. Encapsulated by a high-level API, application-driven service establishment may thus proceed by installing a system service, independent of ISP awareness and without the need of globally upgrading the network.

3 API and Implementation

Multicast application development should be decoupled of technological deployment throughout the infrastructure. It requires a common multicast API that offers calls to transmit and receive multicast data independent of the supporting layer and the underlying technological details. For inter-technology transmissions, a consistent view on multicast states is needed, as well. In contrast to the standard multicast socket interface, the H∀Mcast API abstracts naming and addressing [6]. Using a multicast address in the common socket API predefines the corresponding routing layer. In this approach, the multicast address used for joining a group denotes an application layer data stream that is identified by a multicast URI and without an association to the underlying distribution technology. In addition, a system layer at each device accounts for a *late binding* of names to addresses (i.e., during run-time and not compile-time).

Multicast group names are based on an URI scheme that is defined as follows:

```
scheme "://" group "@" instantiation ":"
             port "/" sec-credentials
```

The *scheme* refers to the namespace of the assigned identifier (e.g., ip, or sip), *group* denotes the group ID, *instantiation* identifies the entity that generates the instance of the group, *port* identifies a specific application, and *sec-credentials* are optional to implement security. Valid group IDs will be ipv://224.0.1.1:4000 or sip://snoopy@peanuts.com, for example. This identifiers are passed to directly to a socket, using high-level group calls.

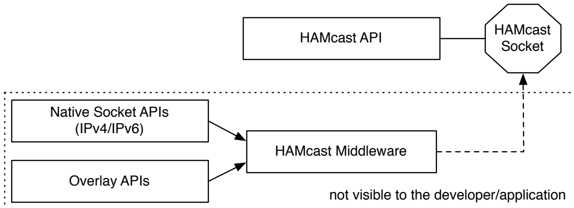

Fig. 2. The H∀Mcast service architecture composed of an abstract socket that grants service access as a facade to one or several network service interfaces via an intelligent middleware

The H∀Mcast socket describes a group communication channel composed of one or multiple interfaces. A socket may be created without explicit interface association by the application, which leaves the choice of the underlying forwarding technology to the group communication stack. However, an application may also bind the socket to one or multiple dedicated interfaces, which predefines the forwarding technology and the namespace(s) of the Group Address(es). All of this happens at a system-layer as visualized in Figure 2 that concentrates hybrid multicast complexity and simplifies application and network duties.

4 Conclusions and Outlook

In this paper, we argued that a system-centric deployment in parts is a promising way to overcome the current Internet feature freeze. For the example of hybrid adaptive multicast, we presented an architecture and prototypic realization of a group communication service that met the requirements by two core contributions. First the various service instantiations and deployments are subsumed below an abstract service entry point implemented as a middleware at end systems. More importantly, a universal service access is granted to applications following a refined general multicast concept and a technology-agnostic API. Thereby and for the first time, group applications can be written once, but run everywhere.

In future work, we will extend implementations and large-scale testing. In particular, support for different programming languages waits to be added. We further plan to intensify standardization work of the API to make these achievements valuable for a wider public.

References

1. Saltzer, J.H., Reed, D.P., Clark, D.D.: End-to-End Arguments in System Design. ACM Trans. Comput. Syst. 2(4), 277–288 (1984)
2. Deering, S.: Host extensions for IP multicasting. IETF, RFC 1112 (August 1989)
3. Schmidt, T., Waehlisch, M., Fairhurst, G.: Multicast Mobility in Mobile IP Version 6 (MIPv6): Problem Statement and Brief Survey. IETF, RFC 5757 (2010)

4. Wählisch, M., Schmidt, T.C.: Between Underlay and Overlay: On Deployable, Efficient, Mobility-agnostic Group Communication Services. Internet Research 17(5), 519–534 (2007)
5. Wählisch, M., Schmidt, T.C.: An a Priori Estimator for the Delay Distribution in Global Hyrid Multicast. In: Proc. of the ACM SIGCOMM CoNEXT.., pp. 19–20. ACM, N.Y (December 2009)
6. Wählisch, M., Schmidt, T.C., Venaas, S.: A Common API for Transparent Hybrid Multicast. individual, IRTF Internet Draft – work in prog. (March 02, 2010)